UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/800,163	03/12/2004	Vijay Deshmukh	5693P048	9062

48102 7590 02/20/2008
NETWORK APPLIANCE/BLAKELY
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
LE, MIRANDA	

ART UNIT	PAPER NUMBER
2167	

MAIL DATE	DELIVERY MODE
02/20/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/800,163	DESHMUKH ET AL.
	Examiner	Art Unit
	MIRANDA LE	2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 January 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18,28,32,34-36,39 and 40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18,28,32,34-36,39 and 40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *01/10/2008*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 11/01/2007 has been entered.

This communication is responsive to Amendment, filed 11/01/2007.

Claims 1-18, 28, 32, 34-36, 39-40 are pending in this application. In the Amendment, claims 19-27, 29-31, 33, 37-38 have been cancelled, claims 1, 10, 28, 32, 34-36 have been amended, claims 39-40 have been added. This action is made non-Final.

Information Disclosure Statement

2. Applicants' Information Disclosure Statement, filed 01/10/2008, has been received, entered into the record, and considered. See attached form PTO-1449.

Claim Objections

3. Claims 32, 34 are objected to because of the following informalities: Claims 32, 34 depend on claim 31 which has been cancelled. Appropriate correction is required. For purposes of examination, claims 32, 34 treated as they are dependent upon claims 28.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 1-18, 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sedlar (US Patent No. 6,922,708), in view of Tan et al. (US Patent No. 6,356,902), and further in view of Chen et al. (US Patent No. 6,625,624).

As per claim 1, Sedlar teaches a method for creating a file information database *(i.e. Emulating other OS File System Characteristics in a Database, col. 10, line 50 to col. 11, line 7)* comprising:

scanning *(i.e. to scan the table, col. 21, lines 38-49)* a storage server *(i.e. database server 204 stores files that originate from numerous distinct OS file systems, col. 12, lines 21-37)* having a directory structure *(i.e. During the traversal of the hierarchical index, ... a child of the directory associated with the directory entry, col. 22, lines 39-47)*;

collecting data regarding the directory structure *(i.e. During the traversal of the hierarchical index, ... a child of the directory associated with the directory entry, col. 22, lines 39-47)*;

for each directory of the directory structure, determining whether *(i.e. Once the index entry 508 for the root directory 610 has been located, the DBMS determines whether there are any more filenames in the input pathname (step 802). If there are no more filenames in the input pathname, then control proceeds to step 820 and the FileID stored in index entry 508 is used to look up the root directory entry in the files table 710, col. 9, lines 10-16)* each member of the directory is a file *(i.e. the FileID of directory 616 ("X3"). As shall be described in greater detail, the information contained in the Dir_entry_list field makes accessing information based on pathnames much faster and easier, col. 8, lines 4-17)* or subdirectory *(i.e. the items that have index entries in the hierarchical index 510 are only those directories that are parents to other directories and/or that are currently storing documents. Those items that do not have children (e.g. Example.doc, Access, App1, App2, App3 of FIG. 6) are preferably not included, col. 7, lines 54-64)*;

assigning a first identification (ID) number to a first determined directory *(i.e. index entry has been created for the Documents directory, col. 23, lines 1-8)* and a second ID number to a second determined directory *(i.e. Word directory, col. 23, lines 9-16)* in the directory structure *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64)*;

examining the determined files *(i.e. Once the index entry 508 for the root directory 610 has been located, the DBMS determines whether there are any more filenames in the input pathname (step 802). If there are no more filenames in the input pathname, then control proceeds to step 820 and the FileID stored in index entry 508 is used to look up the root directory entry in the files table 710, col. 9, lines 10-16)*; and

writing a data structure including the first ID number, the second ID number and relation between the first directory and the second directory *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64).*

Sedlar does not specifically teach:

using first thread to assign a first identification (ID) number to a first determined directory and a second ID number to a second determined directory in the directory structure according to a depth first search (DFS) order;

wherein the directory numbers are assigned while the directory structure is being traversed in the DFS order;

using second thread to examine the determined files.

Tan teaches:

using first thread *(i.e. Parent Stack, See Fig. 2B)* to assign a first identification (ID) number *(i.e. LEV 1...7, Fig. 7)* to a first determined directory and a second ID number *(i.e. LEV 1...7, Fig. 7)* to a second determined directory in the directory structure according to a depth first search (DFS) order *(i.e. means of traversing in depth-first search in a depth coupled graph map without page fault using an efficient traversing algorithm, col. 2, lines 46-59)*;

wherein the directory numbers are assigned while the directory structure is being traversed in the DFS order *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3)*;

using second thread to examine the determined files *(i.e. Child Stack, See Fig. 2B).*

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar and Tan at the time the invention was made to modify the system of Sedlar to include the limitations as taught by Tan. One of ordinary skill in the art would be motivated to make this combination in order to traverse in depth-first search in a depth couple graph map in view of Tan (col. 2, lines 46-59), as doing so would give the added benefit of providing an improved method and system for enhancing data retrieval and storage in a multilevel tree structure by significantly reducing the required amount of time and memory as taught by Tan (col. 2, lines 40-45).

Although Tan teaches the first and second threads as *Parent Stack, Child Stack, See Fig. 2B,* Tan does not state the term "thread".

Chen teaches this limitation at Fig. 2 *(i.e. Agent thread, user thread).*

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar, Tan, Chen at the time the invention was made to modify the system of Sedlar, Tan to include the limitations as taught by Chen. One of ordinary skill in the art would be motivated to make this combination in order to walk the html tree structure to collect and archive pages in view of Chen (col. 3, lines 54-60), as doing so would give the added benefit of enabling users to

retrieve and/or search for old information, even after such information has evolved or disappeared from the original server as taught by Chen (Abstract).

As per claim 10, Sedlar teaches a machine readable medium having stored thereon executable program code which, when executed, causes a machine to perform a method for creating a file information database *(i.e. Emulating other OS File System Characteristics in a Database, col. 10, line 50 to col. 11, line 7)*, the method comprising:

scanning *(i.e. to scan the table, col. 21, lines 38-49)* a storage server *(i.e. database server 204 stores files that originate from numerous distinct OS file systems, col. 12, lines 21-37)* having a directory structure *(i.e. During the traversal of the hierarchical index, ... a child of the directory associated with the directory entry, col. 22, lines 39-47)*;

collecting data regarding the directory structure *(i.e. During the traversal of the hierarchical index, ... a child of the directory associated with the directory entry, col. 22, lines 39-47)*;

for each directory of the directory structure, determining whether *(i.e. Once the index entry 508 for the root directory 610 has been located, the DBMS determines whether there are any more filenames in the input pathname (step 802). If there are no more filenames in the input pathname, then control proceeds to step 820 and the FileID stored in index entry 508 is used to look up the root directory entry in the files table 710, col. 9, lines 10-16)* each member of the directory is a file *(i.e. the FileID of directory 616 ("X3"). As shall be described in greater detail, the information contained in the Dir_entry_list field makes accessing information based on pathnames much faster and easier, col. 8, lines 4-17)* or subdirectory *(i.e. the items that have*

index entries in the hierarchical index 510 are only those directories that are parents to other directories and/or that are currently storing documents. Those items that do not have children (e.g. Example.doc, Access, App1, App2, App3 of FIG. 6) are preferably not included, col. 7, lines 54-64);

assigning a first identification (ID) number to a first determined directory *(i.e. index entry has been created for the Documents directory, col. 23, lines 1-8)* and a second ID number to a second determined directory *(i.e. Word directory, col. 23, lines 9-16)* in the directory structure *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64)*;

examining the determined files *(i.e. Once the index entry 508 for the root directory 610 has been located, the DBMS determines whether there are any more filenames in the input pathname (step 802). If there are no more filenames in the input pathname, then control proceeds to step 820 and the FileID stored in index entry 508 is used to look up the root directory entry in the files table 710, col. 9, lines 10-16)*; and

writing a data structure including the first ID number, the second ID number and a relation between the first directory and the second directory *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64).*

Sedlar does not specifically teach:

using first thread to assign a first identification (ID) number to a first determined directory and a second ID number to a second determined directory in the directory structure according to a depth first search (DFS) order;

wherein the directory numbers are assigned while the directory structure is being traversed in the DFS order;

using second thread to examine the determined files.

Tan teaches:

using first thread *(i.e. Parent Stack, See Fig. 2B)* to assign a first identification (ID) number *(i.e. LEV 1…7, Fig. 7)* to a first determined directory and a second ID number *(i.e. LEV 1…7, Fig. 7)* to a second determined directory in the directory structure according to a depth first search (DFS) order *(i.e. means of traversing in depth-first search in a depth coupled graph map without page fault using an efficient traversing algorithm, col. 2, lines 46-59)*;

wherein the directory numbers are assigned while the directory structure is being traversed in the DFS order *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3)*;

using second thread to examine the determined files *(i.e. Child Stack, See Fig. 2B).*

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar and Tan at the time the invention was made to modify the system of Sedlar to include the limitations as taught by Tan. One of ordinary skill in the art would be motivated to make this combination in order to traverse in depth-first search in a depth couple graph map in view of Tan

(col. 2, lines 46-59), as doing so would give the added benefit of providing an improved method and system for enhancing data retrieval and storage in a multilevel tree structure by significantly reducing the required amount of time and memory as taught by Tan (col. 2, lines 40-45).

Although Tan teaches the first and second threads as *Parent Stack, Child Stack, See Fig. 2B,* Tan does not state the term "thread".

Chen teaches this limitation at Fig. 2 *(i.e. Agent thread, user thread).*

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar, Tan, Chen at the time the invention was made to modify the system of Sedlar, Tan to include the limitations as taught by Chen. One of ordinary skill in the art would be motivated to make this combination in order to walk the html tree structure to collect and archive pages in view of Chen (col. 3, lines 54-60), as doing so would give the added benefit of enabling users to retrieve and/or search for old information, even after such information has evolved or disappeared from the original server as taught by Chen (Abstract).

As per claim 39, Sedlar teaches a method for creating a file information database *(i.e. Emulating other OS File System Characteristics in a Database, col. 10, line 50 to col. 11, line 7)* comprising:

scanning *(i.e. to scan the table, col. 21, lines 38-49)* a storage server *(i.e. database server 204 stores files that originate from numerous distinct OS file systems, col. 12, lines 21-37)* having a directory structure *(i.e. During the traversal of the hierarchical index, ... a child of the directory associated with the directory entry, col. 22, lines 39-47)*;

for each directory of the directory structure, determining whether *(i.e. Once the index entry 508 for the root directory 610 has been located, the DBMS determines whether there are any more filenames in the input pathname (step 802). If there are no more filenames in the input pathname, then control proceeds to step 820 and the FileID stored in index entry 508 is used to look up the root directory entry in the files table 710, col. 9, lines 10-16)* each member of the directory is a file *(i.e. the FileID of directory 616 ("X3"). As shall be described in greater detail, the information contained in the Dir_entry_list field makes accessing information based on pathnames much faster and easier, col. 8, lines 4-17)* or subdirectory *(i.e. the items that have index entries in the hierarchical index 510 are only those directories that are parents to other directories and/or that are currently storing documents. Those items that do not have children (e.g. Example.doc, Access, App1, App2, App3 of FIG. 6) are preferably not included, col. 7, lines 54-64)*;

assigning a first identification (ID) number to a first determined directory *(i.e. index entry has been created for the Documents directory, col. 23, lines 1-8)* and a second ID number to a second determined directory *(i.e. Word directory, col. 23, lines 9-16)* in the directory structure *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64)*;

examining the determined files *(i.e. Once the index entry 508 for the root directory 610 has been located, the DBMS determines whether there are any more filenames in the input pathname (step 802). If there are no more filenames in the input pathname, then control*

proceeds to step 820 and the FileID stored in index entry 508 is used to look up the root directory entry in the files table 710, col. 9, lines 10-16); and

writing a data structure including the first ID number, the second ID number and relation between the first directory and the second directory *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64).*

Sedlar does not specifically teach:

using first thread to assign a first identification (ID) number to a first determined directory and a second ID number to a second determined directory in the directory structure according to a depth first search (DFS) order;

wherein the directory numbers are chronologically assigned in numerical order while the directory structure is being traversed in the DFS order;

using second thread to examine the determined files.

Tan teaches:

using first thread *(i.e. Parent Stack, See Fig. 2B)* to assign a first identification (ID) number *(i.e. LEV 1 ... 7, Fig. 7)* to a first determined directory and a second ID number *(i.e. LEV 1 ... 7, Fig. 7)* to a second determined directory in the directory structure according to a depth first search (DFS) order *(i.e. means of traversing in depth-first search in a depth coupled graph map without page fault using an efficient traversing algorithm, col. 2, lines 46-59)*;

wherein the directory numbers are chronologically assigned while the directory structure is being traversed in the DFS order *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack.*

Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3);

using second thread to examine the determined files *(i.e. Child Stack, See Fig. 2B).*

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar and Tan at the time the invention was made to modify the system of Sedlar to include the limitations as taught by Tan. One of ordinary skill in the art would be motivated to make this combination in order to traverse in depth-first search in a depth couple graph map in view of Tan (col. 2, lines 46-59), as doing so would give the added benefit of providing an improved method and system for enhancing data retrieval and storage in a multilevel tree structure by significantly reducing the required amount of time and memory as taught by Tan (col. 2, lines 40-45).

Tan teaches the first and second threads as *Parent Stack, Child Stack, See Fig. 2B.*

Tan does not state the term "thread".

Chen teaches this limitation at Fig. 2 *(i.e. Agent thread, user thread).*

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar, Tan, Chen at the time the invention was made to modify the system of Sedlar, Tan to include the limitations as taught by Chen. One of ordinary skill in the art would be motivated to make this combination in order to walk the html tree structure to collect and archive pages in view of Chen (col. 3, lines 54-60), as doing so would give the added benefit of enabling users to retrieve and/or search for old information, even after such information has evolved or disappeared from the original server as taught by Chen (Abstract).

As to claims 2, 11, Tan teaches the agent has a first file system (i.e. Service GateWay Servers Entity, See Fig. 1), and the scanning and collecting by using an agent separate from the storage server (i.e. Content Provider Entity, See Fig. 1).

As to claims 3, 12, Tan teaches the agent has a first file system (i.e. Service GateWay Servers Entity, See Fig. 1), and the storage server has a second file system (i.e. Content Provider Entity, See Fig. 1), and wherein the first file system is different from the second file system (See Fig. 1).

As to claims 4, 13, Sedlar teaches the relation indicates that the first directory is an immediate child of the second directory *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64).*

As to claims 5, 14, Tan teaches assigning further comprises assigning the ID number *(i.e. LEV 1...7, Fig. 7)* while colleting the data *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3).*

As to claims 6, 15, Sedlar teaches writing the data structure *(i.e. an index entry for the Document directory is added to the hierarchical index, ... the Dir_Entry_List is updated to*

indicate that the new Document directory is a child of the Word directory, col. 22, lines 56-64) further comprises writing the data structure to a database server *(i.e. Database server, Fig. 3).*

As to claims 7, 16, Sedlar teaches:

receiving a request to determine the parent of the first directory *(i.e. the pathname resolution process for locating a file within an emulated file system begins by locating the index entry 508 of the root directory 610 (step 800), col. 8, line 56 to col. 9, line 9)*; and

referencing the relation between the first directory and the second directory of the data structure to determine the parent of the first directory *(i.e. the pathname resolution process for locating a file within an emulated file system begins by locating the index entry 508 of the root directory 610 (step 800). Because all pathname resolution operations begin by accessing the root directory's index entry 508, data that indicates the location of the index entry for the root directory 610 (index entry 508) may be maintained at a convenient location outside of the hierarchical index 510 in order to quickly locate the index entry 508 of the root directory at the start of every search, col. 8, line 56 to col. 9, line 9).*

As to claims 8, 17, Sedlar teaches:

receiving a request to determine an immediate child of the second directory *(i.e. Consulting the Dir_entry_list of index entry 512, the system searches for the next filename in the input pathname (steps 804 and 806). In the present example, the filename "Word" follows the filename "Windows" in the input pathname, col. 9, line 56 to col. 10, line 7);*

searching the data structure to find any relation, including the relation between the first directory and the second directory, which indicates that the second directory is a parent in said relation *(i.e. the system searches the Dir_entry_list of index entry 512 for an array entry for "Word", col. 9, line 56 to col. 10, line 7);* and

determine the immediate child of the second directory based on said any relation *(i.e. Since Word directory 616 is just part of the specified path and not the target, files table 710 is not consulted. Instead, the system uses the RowID (Y3) to locate the index entry 514 for Word directory 616 (step 824), col. 9, line 56 to col. 10, line 7).*

As to claims 9, 18, Tan teaches:

receiving a request to determine a set of ID number *(i.e. LEV 1...7, Fig. 7)* of every child of a third directory in the directory structure, wherein the third directory is assigned a third ID number *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3)*;

determining fourth ID number *(i.e. LEV 1...7, Fig. 7)* of a sibling of the third directory*(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3)*; and

determining the set of ID number *(i.e. LEV 1...7, Fig. 7)* between the third ID number and the fourth ID number *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3).*

6. Claim 40 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sedlar (US Patent No. 6,922,708), in view of Tan et al. (US Patent No. 6,356,902), in view of Chen et al. (US Patent No. 6,625,624), and further in view of Gasser et al. (US Patent No. 6,636,250).

As per claim 40, Sedlar, Tan, Chen do not explicitly teach a top level directory of the directory structure is assigned an ID of "0" (zero).

Gasser teaches this limitation in Fig. 9.

It would have been obvious to one of ordinary skill of the art having the teaching of Sedlar, Tan, Chen, Gasser at the time the invention was made to modify the system of Sedlar, Tan, Chen to include the limitations as taught by Gasser. One of ordinary skill in the art would be motivated to make this combination in order define in a first level of the tree serve as group arrangement descriptors in view of Gasser (col. 5, lines 30-57), as doing so would give the added benefit of allowing a user to overlay multiple relationships on top of one another so that the user can determine and view the relationships at the same time in the same view of the graphical user interface as taught by Gasser (col. 2, line 48 to col. 8, line 9).

7. Claims 28, 34-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al. (US Patent No. 6,625,624), and further in view of Tan et al. (US Patent No. 6,356,902).

As per claim 28, Chen teaches a method for creating a logical tree comprising:

using a directory walking thread *(i.e. walking thread, See Fig. 2)* to examine a first directory *(i.e. a root URL, col. 7, lines 20-32)* from a top of a directory queue *(i.e. As described above, the walking can be defined by a root URL and parameterized by (a) the depth of walking through hyper-references under HTML pages, (b) with or without image files embedded in HTML pages, and (c) walking through pages on the local web site or on all web sites, col. 7, lines 20-32)* and determine a set of children of the directory *(i.e. archive pages, col. 3, lines 54-60; The iagent class 2035 of the agent thread 2030 connects to a remote web server 70 or proxy to request a Web page. iagent 2035 can cache the page and return the page to iserver 2025 or to ihtwalk 2045, a facility to walk the html tree structure to collect and archive pages. The ihtwalk class facility is further described below in Section 4 in the description of the walking facilities, col. 3, lines 54-60)*;

examining a set of children *(i.e. HTML pages, col. 7, lines 20-32)* of the first directory to determine a first subset of files and a second subset of directories *(i.e. walking through pages on the local web site or on all web sites, col. 7, lines 20-32)*;

placing the first subset of files in a file queue for examination by a file thread *(i.e. the walking facility may be used to visit the sets of web pages that are designated to be packed, col. 8, lines 3-23)*; and

placing the second subset on the top of the directory queue *(i.e. The system then walks through the set of web pages rooted by the designated URL and packs them into a package using the designated name col. 8, lines 3-23).*

Chen does not specifically teach:

assigning a depth first search (DFS) ID to the first directory, wherein the directory numbers are assigned while the directory structure is being traversed in the DFS order.

Tan teaches:

assigning a depth first search (DFS) ID to the first directory *(i.e. means of traversing in depth-first search in a depth coupled graph map without page fault using an efficient traversing algorithm, col. 2, lines 46-59)*, wherein the directory numbers are assigned while the directory structure is being traversed in the DFS order *(i.e. For each level of the tree structure, the graph nodes are stored in the Parent-stack, the adjacent level of tree structure is stored in the child-stack. Assignment of the link pointer of the graph node in the parent-stack is stated in the conditions shown in step 2.9, col. 5, line 7 to col. 6, line 3).*

It would have been obvious to one of ordinary skill of the art having the teaching of Chen and Tan at the time the invention was made to modify the system of Chen to include the limitations as taught by Tan. One of ordinary skill in the art would be motivated to make this combination in order to traverse in depth-first search in a depth couple graph map in view of Tan (col. 2, lines 46-59), as doing so would give the added benefit of providing an improved method and system for enhancing data retrieval and storage in a multilevel tree structure by significantly reducing the required amount of time and memory as taught by Tan (col. 2, lines 40-45).

As per claim 34, Chen teaches the directory walking thread *(i.e. walking thread, See Fig. 2)* is hosted by an agent *(i.e. Agent Thread, See Fig. 2)* that is separate from the storage server *(See Fig. 1).*

As per claim 35, Chen teaches using an MMA *(i.e. IProxy main thread 2010, Fig. 2)* to control the agent *(See Figs. 1, 2).*

As per claim 36, Chen teaches the directories are hosted by is a filer *(i.e. Agent Thread, See Fig. 2).*

8. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al. (US Patent No. 6,625,624), in view of Tan et al. (US Patent No. 6,356,902), and further in view of Sedlar (US Patent No. 6,922,708).

As per claim 32, Chen and Tan do not specifically teach examining the file queue further comprises recording information about a first file taken from the file queue.

Sedlar teaches wherein examining the file queue further comprises recording information about a first file taken from the file queue *(i.e. an event server executing external to a database server is registered as a subscriber to a queue managed by the database server. The queue to which the event server subscribes shall be referred to herein as the file event queue. Entities that are interested in particular file system events register their interest with the event server. The*

event server communicates with the database server through the database API, and with the interested entities through the protocols supported by those entities, col. 28, lines 39-47).

It would have been obvious to one of ordinary skill of the art having the teaching of Chen, Tan, Sedlar at the time the invention was made to modify the system of Chen, Tan to include the limitations as taught by Sedlar. One of ordinary skill in the art would be motivated to make this combination in order to communicate with the database server in view of Sedlar (col. 28, lines 39-47), as doing so would give the added benefit of obtaining the process of formulating and submitting queries to a database server is less intuitive than merely traversing a hierarchy of directories, and is beyond the technical comfort level of many computer users as taught by Sedlar (col. 3, lines 4-12).

Response to Arguments

9. Applicant's arguments regarding the newly amended claims have overcome the cited arts, with respect to claims 1-18, 28, 32, 34-36, 39-40 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Miranda Le whose telephone number is (571) 272-4112. The examiner can normally be reached on Monday through Friday from 8:30 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham, can be reached on (571) 272-7079. The fax number to this Art Unit is (571)-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Miranda Le

Miranda Le
February 15, 2008